IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 7 February 2025, accepted 24 February 2025, date of publication 27 February 2025, date of current version 7 March 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3546311

METHODS

Remote Real-Time Testing of Physical Components Using Communication Setup Automation

ALEXANDRA BACH[1] **AND ANTONELLO MONTI**[1,2], **(Senior Member, IEEE)**
[1]Institute for Automation of Complex Power Systems, RWTH Aachen University, 52074 Aachen, Germany
[2]Fraunhofer Institute for Applied Information Technology, 52074 Aachen, Germany

Corresponding author: Alexandra Bach (alexandra.bach@eonerc.rwth-aachen.de)

This work was supported by the European Union's Horizon 2020 Research and Innovation Program through the Project HYPERRIDE under Grant 957788.

ABSTRACT This paper addresses the challenge of testing physical components in remote grids. It is a critical area often constrained by limitations of existing real-time simulation-based methods. Current literature emphasizes real-time simulators using analog input and output interfaces for safe testing. However, this approach is hindered by high investment cost and necessary operational knowledge. In contrast, our method focuses solely on a component-specific expertise. This offers a more accessible and feasible testing environment. Despite its potential, the pure physical component approach is discussed barely in current literature. Particularly, when it comes to testing complex grid systems. To overcome this obstacles, we introduce a innovative solution: remote component testing distributed across several facilities. This paper outlines a robust architecture. It integrates power converters and the virtual gateway VILLASnode. This enables seamless connectivity between remote grids via the internet and addresses the shortcomings of traditional single facility testing environments. Furthermore, we include an automated communication setup along with a comprehensive delay analysis to ensure that real-time requirements are met effectively. A comparative study of common communication protocols is included to guide future implementations. To validate our architecture's functionality, we conduct a case study that emulates unavailable remote physical components in a microgrid using two different control strategies. This successfully demonstrates the exchange and emulation of power setpoints. Our findings reveal facility-specific limitations - such as resolution constraints - that persists regardless of architectural advancements. This paper lays a solid foundation for remote physical component testing. It paves the way for replication and further exploration through more sophisticated case studies in the future.

INDEX TERMS Communication automation, geographically distributed facilities, real-time experiment, remote coupling, remote physical component testing.

I. INTRODUCTION

The growing demand for sustainable energy is transforming the traditional power grid. As the power system becomes more complex and expansive, researchers need approaches to large-scale experiments. These experiments are necessary to analyze and validate its interactions with new technologies. Testing the integration of hardware and software components, including communication, presents new challenges.

Different approaches exist to test individual physical components on device and system level. Device level concentrates on the device itself, while the system level investigates the interaction of the device with the system. This paper distinguishes between two categories: a real-time simulator approach on device level, which facilitates interaction between simulated grids and physical components, and a purely physical component approach on device or system

The associate editor coordinating the review of this manuscript and approving it for publication was Inam Nutkani.

level, which enables interaction only between physical components and real grids.

Real-time simulation means that the processes in the simulation are run in a pre-defined time frame. Particularly for power system simulations, it means that one time step in the simulation is equal to one time step in the real world [1]. The advantage of a real-time simulator is the simulation of a grid so that components are safely tested using analog input and output interfaces, also known as power hardware-in-the-loop (HiL) [1]. The authors of [2] demonstrate the feasibility of large-scale implementation of this approach by coupling real-time simulators of different vendors analyzing control schemes and devices using the virtual gateway VILLASnode [3]. The experiment reported in [2] is distributed over several laboratories using available simulations and physical systems. The benefit is that the testing is realized in a fully or nearly fully controlled environment. The limitation is the use of real-time simulators because they are expensive, only available in some facilities, and require training to be used. Moreover, a component can only be tested on device level.

The physical-component-only approach uses physical components to emulate a real grid as experimental environment. The advantage of testing in a real grid is that the components are no longer in a controlled environment. They need to react to unforeseen real events. A testing on device and system level is possible. The authors of [4] present the local testing of converters in an emulated grid. The emulated grid is realized by power electronics converters. They overcome the limitations of real-time simulators, e.g., less stability due to transmission delays. However, the authors of [4] mention that there is no method yet to emulate complex grids. This refers mainly to the available size of the local grid, such as the number of converters, or training of personnel. Originating from the real-time simulator approach as in [2], this paper proposes remote testing of components and grids distributed over several facilities. The aim is to overcome the limitations in a single facility testing environment.

Given the complexity of the testing environments above, the automation of setting up of the communication between the components and distributed facilities is a challenge in itself. Automation is needed to make the testing more user friendly and to avoid the repetition of errors in the setup. Usually, researchers perform testing who are not specialized on communications. Testing needs communication between the physical components in remote grids but can also be extended to normal operation in the future, e.g., to advance existing control. The idea of the communication setup is that it should be as light as possible to run it on nearly every component. Web Real-Time Communication protocol (WebRTC) is suitable for automated communication setup. It is a widely used protocol for real-time video calls [5]. It ensures quick and automated communication setup and reliable data exchange over the internet. So far, a common protocol to use in a distributed experiment is User Datagram Protocol (UDP) [2]. The main limitation is that a lot of firewalls block UDP per default. The authors of [6] suggest the use of WebRTC for distributed experiments and provide a study of different performance evaluations. However, this study misses the level of automation introduced by WebRTC as well as a direct comparison with UDP under constrained and unreliable communication channels like the internet.

In distributed experiments, real-time data exchange is fundamental to ensure that the experiment has a high fidelity in terms of latencies. Some publications focus on the derivation of delay models based on delay measurements. The authors of [7] analyze the delay and derive the different contributions for component testing in real-time simulator approach. This analysis is specific for the gateway used in [7]. Other real-time experiments like [2] or [6] use the virtual gateway VILLASnode to couple components which is also used in this paper. However, an analysis of the underlying virtual gateway itself and an analysis of the network interfaces is missing to derive the impact on the real-time behavior.

This paper aims to provide advancements in physical component testing in remote grids including communication automation. The provided architecture is designed independent from the underlying system. It needs converters and the virtual gateway VILLASnode. This paper addresses the previous challenges by the following key contributions:

1) A lightweight architecture on top of existing systems in terms of ease of installation and use.
2) Demonstration of the performance of automation of the communication setup by using WebRTC to couple components. The indicator is the number of required user interactions and the invested time which are both reduced significantly.
3) A performance comparison between UDP and WebRTC under unreliable communication channel conditions. Results show that WebRTC is faster indicated by round-trip time and thus a very good enabler to send control signals over the internet.
4) A delay analysis model of the virtual gateway in terms of network interface and conversion time. Results show in detail the different delay contributions indicated by various latency measurements.
5) Demonstration of the functionality of the architecture in form of a case study.

The remainder of this paper is structured as followed: first the architecture is described in more detail explaining the requirements of converters. The modularity and interoperability of VILLASnode is stressed and the automation of WebRTC is assessed. Then, a delay analysis model is derived. It provides delay measurements introduced by the virtual gateway VILLASnode and the network interfaces. This section also provides a comparison between WebRTC and UDP. Section IV introduces the different facilities and points out how they challenge the architecture in form of a case study. It provides an application for possible use of the architecture including limitations. Lastly, conclusions are drawn and an outlook is provided.

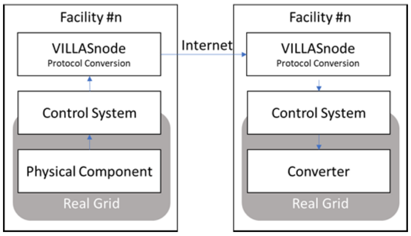

FIGURE 1. General architecture based on VILLASnode and converters.

II. ARCHITECTURE

The design of the architecture for physical component testing in remote grids has the following objectives: modularity of the architecture, interoperability between the components, emulation of physical components in remote grids, and communication setup automation. The architecture is designed on top of existing systems and interacts between remote grids independent from the system. The architecture concentrates on evaluation of physical component integration and communication between different facilities on system level. This would not be possible in one single facility on device level. This section derives the requirements from these objectives.

Usually, as described in [1], electrical or control signals are exchanged between the remote facilities. However, they are affected by transmission delays of the internet. For instance, AC electrical signals have a phase that would change when adding the internet delay. In case of a control signals on device level, where the dynamics are in range of kilohertz, a remote testing is not even possible because the internet latency cannot fulfill these requirements. For this reason, the use of power setpoints as control signal coupling on system level is proposed. Devices and grids can still exchange values with each other, but with less time constraints. It is a common procedure in system control, e.g. as described by the authors of [9].

Figure 1 shows the general architecture including a real grid and a VILLASnode instance each for one geographically distributed facility, i.e., distinct and separated locations which are not connected physically. The goal is to emulate the physical component from one facility in another facility via a converter (see Section II-A) and to exchange data by VILLASnode (see Section II-B).

First, the data of the physical component is extracted via the control system. The control system controls the physical components by sending or receiving power setpoints.

The setpoints are forwarded to the local VILLASnode instance. It runs in the same facility, thus the same network, on top of the control system, e.g., in a dedicated Linux machine or a Raspberry Pi. It can reach the control system by using the same protocol as the control system.

VILLASnode converts the protocol of the control system to WebRTC which is the protocol to talk to the other remote VILLASnode instance and is optimized for internet transmissions. VILLASnode forwards the data to another remote facility reachable via the internet. In the remote facility, the setpoint included in WebRTC is changed to the required local protocol. Then, the setpoint is forwarded to the converter via the control system so that the physical component is emulated by the converter. In this way, a physical component can be tested in a remote grid.

This architecture is lightweight because it works with the existing facility and only requires the installation of VILLASnode as Docker container. It runs in the same facility as the real grid and can interact with it via the control system. Moreover, the communication setup is very user friendly because it does not use specific requirements from the facility side like security adaptions. These points are explained in more detail in the following sub-sections.

TABLE 1. Requirements for emulation in remote grids.

Architecture	Design
Coupling Technique	Setpoints
Control System Communication Protocols	UDP, TCP, Modbus, IEC 61850, IEC 60870 (generally supported VILLASnode protocols, see [3])
Control System Safety	Read and write (send and receive setpoints)
Physical Component	Converter, inverter, switch gear, transformer, BESS
Frequency of Setpoint Updates for Real-time	Time between updates > overall delay
Power Range	Remote Converter max. power $\geq$ Physical Component max. power (if not fulfilled, include safety mechanisms in VILLASnode)

A. EMULATION IN REMOTE GRID WITH REMOTE CONVERTER

A remote converter is a converter integrated in a remote grid to emulate a physical component. The converter serves as physical interface to the remote grid on system level as shown in Figure 1. The physical component to be emulated requires an interface to a control system. A control system is available in nearly all grids because it facilitates the monitoring and management. The control system can reach the physical component and can send (write) or receive (read) setpoints. Based on the implementation of the control system, the communication protocol changes. Typical protocols are UDP, TCP, Modbus, or IEC 61850 and IEC 60870 which are supported by VILLASnode.

The physical component itself can be a converter, inverter, switch gear, transformer, or a battery. Other components which also fulfill the above-mentioned requirements are also possible. The physical components require periodic or on demand setpoint updates depending on the facility or the operation. The time between the updates should be at least in the same range as the sum of internet latency and the delay of the architecture itself, short the overall delay, so that the experiment runs in real-time. Otherwise, the behavior of the physical component in the remote grid cannot be emulated in a feasible way. Table 1 gives an overview of these discussed requirements.

Since the remote converter emulates the received setpoints and thus the physical component, the power range of the remote converter and the remote grid need to fit to the power the physical component needs to exchange. It means that the setpoints from the physical component match with the remote converter and remote grid and vice versa. This needs to be checked before the experiment and depends on the individual facility. The power range and the dynamics change depending on the facility, devices, and scenarios. It is recommended to adapt VILLASnode with a safety mechanism by checking the setpoints and adapt them, e.g., by scaling them or cutting the peaks.

B. MODULARITY AND INTEROPERABILITY USING VILLASnode

VILLASnode is a real-time open-source software tool designed to enable geographically distributed experiments. It runs as installation from source or as Docker container [3]. VILLASnode is an interface which enables the coupling of local and remote components without specific requirements. It supports a modular and flexible architecture design based on a configuration file. Moreover, it takes care of protocol conversions and data format adaptions.

These characteristics are very useful for the architecture described here. VILLASnode supports the creation of the architecture by providing every component a set of protocols to link between the local and remote grid and thus provides interoperability. It is used as Docker container which makes the use and the installation very user friendly. It supports the modularity of the setup because every component and grid can be integrated and can interact with the control system. It means that the modularity and interoperability of the architecture is only limited by the number of supported protocols of VILLASnode.

C. COMMUNICATION AUTOMATION USING WEBRTC

VILLASnode uses web real-time communication (WebRTC) between the facilities to exchange data. It is an application layer protocol which uses UDP or TCP as transport. It takes care of Network Address Translation (NAT) traversal, encryption, and peer-to-peer data handling. WebRTC can establish connections under constrained conditions, e.g., from behind firewalls. It does not need to exchange IP addresses but is based on session identifiers. Thus, it requires less support of IT departments, e.g., opening specific ports or allowing IP addresses is usually not necessary. A virtual private network (VPN) is also not required. Since no extra configurations are needed from user side except for the typical VILLASnode configuration, WebRTC introduces automation in the experiment. This level of automation can be measured by the required user interactions as shown in Table 2 compared with a traditional VPN setup.

In this architecture, VILLASnode receives the setpoints from the control system in a specific protocol and format, converts it to WebRTC and sends it to the remote facility where the same steps are undertaken reversely.

TABLE 2. Requirements for emulation in remote grids.

WebRTC	VPN (e.g., via UDP)
Start VILLASnode ⇒ Total user interactions: 1 ⇒ Estimated duration: 1 s	Apply for VPN Install VPN Check with IT department (configure firewalls) Start VILLASnode ⇒ Total user interactions: 4 ⇒ Estimated duration: 1-4 week(s)

D. LIMITATIONS

The proposed architecture might face some limitations regarding its scalability, security during data transmission and performance regarding network reliability.

VILLASnode is designed to be scalable allowing it to handle additional users dynamically. However, as the number of participants scales, the complexity of managing them increases. Large-scale deployments may require additional tools to facilitate this process. The highest reported number of VILLASnode instances is 14 [2]. This was still manageable, but future experiments might include up to 50 instances, e.g., to establish a joint distributed experiment platform.

WebRTC incorporates robust security features, including end-to-end encryption of data streams such as DTLS (Datagram Transport Layer Security) and Stream Control Transmission Protocol (SCTP). Despite these protections, vulnerabilities can still arise. For instance, unauthorized user could join a session or the real IP addresses are exposed. The implementation in VILLASnode takes care of the first point by using session identifiers which are only known to the two participants. The latter point is addressed by hosting the NAT traversal at RWTH.

The system's performance can be affected by poor network conditions, such as high latency, packet loss, or low bandwidth. WebRTC's built-in congestion control mechanisms can help mitigate some of these issues by dynamically adjusting the quality of the data streams. However, WebRTC still relies on the quality of the underlying internet connection. Generally, a good internet connection is the basis for geographically distributed experiments which needs to be ensured before-hand.

III. DELAY ANALYSIS MODEL

Time delay is a major challenge in real-time experiments to ensure high fidelity. In this paper, the delay impacts when the remote converter and the remote grid receive the setpoints from the physical component. The exact requirements are based on the application of the architecture, e.g., grid stability or load control. In worst case, the delay is too large and the setpoints are received too late so that the remote grid cannot operate safely.

This section assesses the introduced delays of the virtual gateway and network interfaces in form of a delay analysis model. It should serve as comparison for future tests. The indicators are latency measurements. The results presented here show three outcomes: effectiveness of WebRTC in

TABLE 3. Software architecture for testing.

Distribution	Linux Ubuntu 22.04
Architecture	Intel(R) Xeon(R) CPU E5-2643 v4 @ 3.40GHz

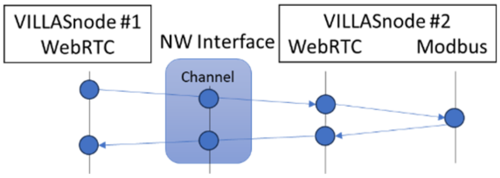

FIGURE 2. General architecture based on VILLASnode and converters.

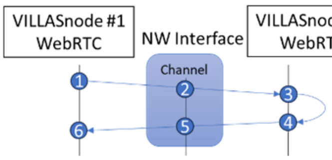

FIGURE 3. RTT contributions.

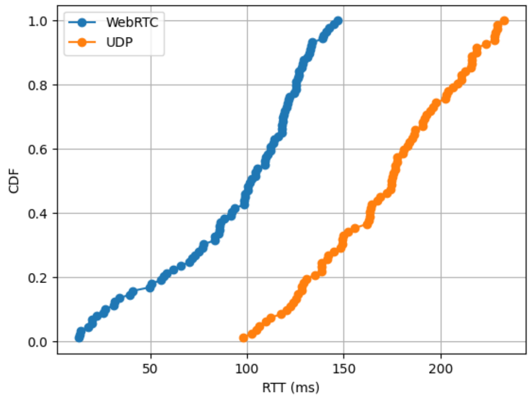

FIGURE 4. Comparison WebRTC and UDP with network emulation.

contrast to UDP in noisy channels, latency of network interfaces, and delay of virtual gateway in form of protocol conversion time. The results are expected to be in microsecond range to provide a very fast architecture.

The software architecture of the system is depicted in Table 3: a Linux system with Ubuntu distribution is used with two central processing units (CPUs). For the test, the platform consists of two VILLASnode instances. Each one runs as a Docker container. Both instances have a WebRTC interface to talk to each other. One instance has an additional Modbus interface so that an exemplary protocol conversion time is calculated. They talk to each other over a network (NW) interface which can emulate delay and packet loss.

Figure 2 shows the different delay contributions in form of sequence diagram. It is differentiated between the VILLASnode instance numbers and their node-types WebRTC and Modbus as well as the NW interface. The blue circles symbolize an uncertainty which is introduced to the unknown reaction of the underlying system itself to commands. Some of them also serve as measurement points. The arrows symbolize the data transfer. This kind of delay analysis allows to establish precise boundaries how long specific transmissions take which help future replicated experiments to compare the results. The following sub-sections derive the delays.

A. ROUND-TRIP TIME (RTT) CONTRIBUTIONS

Different measurements are taken to investigate different contributions of RTT. Figure 3 shows the measurement points and the involved VILLASnode instances and node-types. The first measurements are taken between point 2 and 5. Point 2 takes the timestamp when an application data packet from the first VILLASnode instance is sent out. Point 5 takes the timestamp when the loop backed response is received from the second instance. These two values are subtracted so that the time is calculated it takes for the Linux machine to forward the packet from the interface to the Docker container running VILLASnode, process the packet in VILLASnode to loop back a response as well as for the interface to receive the response. The average time is 0.17 ms.

Other measurements were conducted between point 1 and 3. The difference is the time it takes so that a packet is sent and received in VILLASnode, short the one-way RTT. The average time was 0.24 ms. Measurements between point 1 and 6 are the overall RTT which takes 0.48 ms in average. It is double of the one-way RTT which leads to the assumption that both directions have the same characteristic. The processing time at the second VILLASnode instance is neglectable according to the formula:

$$T_{RTT} = 2 \cdot T_{OneWay} + T_{Processing}$$

However, neglectable does not mean that the processing time is zero. It means that in this case it could not be measured and is probably included in the one-way RTT.

Excluding the average time between point 2 and 5 and assuming that both directions have the same delay contribution, it takes 0.155 ms between VILLASnode generating a packet and the NW interface sending out the packet. Since the time between point 2 and 3 is approximately 0.085 ms, it is very likely that 0.07 ms are caused by the NW interface.

B. COMPARISON WEBRTC AND UDP IN NOISY CHANNELS

This section tests the characteristics of WebRTC and UDP for a communication channel with latency. This direct comparison under latency is not done yet in literature. For the comparison, the same test as in Section III-A between measurement point 2 and 5 is repeated with the Linux internal network emulation (netem). Netem is also included in VILLASnode and introduces an extra latency of 100 ms, with a 30% variation of ± 10 ms. Figure 4 shows the results as cumulative distribution function (CDF). It shows the RTT in milliseconds on the x-axis and the probability that a measurement is smaller or equal than this RTT. 50% of

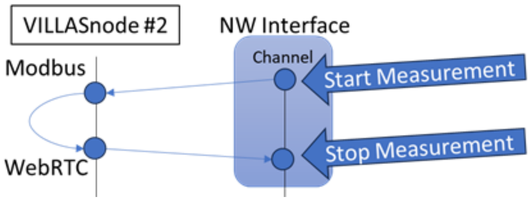

FIGURE 5. Protocol conversion time.

the WebRTC values are equal or smaller than 100 ms. This is equal to the configured latency, i.e., that WebRTC does not introduce additional delays, but follows the latency of the channel. UDP shows a worse performance. Although the distribution has the same width, it starts at 100 ms and shows larger values. It seems that it cannot cope efficiently with the delay. Thus, WebRTC has a more effective performance than UDP regarding delays.

A disadvantage of WebRTC might occur if a direct connection is not possible and a relay sever needs to be used. Using a relay server introduces additional latency because data needs to be routed over additional hops. Setting up a VPN avoids the use of a relay server but increases the setup time, e.g., by including firewall changes. Also the use of different transport protocols impacts the performance. If no relay server is used and a connection can be established directly, UDP might be preferred. UDP is a transport protocol and has less overhead than WebRTC. It is ideal for low latency applications because it provides direct packet delivery without retransmissions. It is usually applied when the participants are in the same network. Distributed experiments can use UDP in combination with a VPN. However, it has the same disadvantages as described before. Moreover, UDP does not support an automated communication setup as described in Section II-C. The authors of [6] provide measurements and results regarding this comparison without considering noisy channels.

In general, the use of WebRTC or UDP depends on the experiment and is always a trade-off. On the one hand, the required time to setup the communication of the experiment needs to be assessed. If it is a short experiment, then the needed time to set up a VPN and change firewall configurations are not worth it. On the other hand, the routing needs to be considered and the additional delay introduced by it. In this paper, a relay server was not used and the routing did not affect the latency. Moreover, the connection between the facilities was only valid for a limited time because the physical components were monitored by personnel during the experiments.

C. PROTOCOL CONVERSION TIME

Figure 5 shows in detail how the protocol conversion time was measured. The first timestamp is taken at the NW interface when the Modbus packet comes from the Modbus server. Then, the packet is converted from Modbus to WebRTC in VILLASnode. The second timestamp is taken when the interface receives the WebRTC packet. Both timestamps are subtracted. The average time is 0.42 ms. This difference includes the time from the interface to VILLASnode and vice versa. Respecting the previous measurement of 0.17 ms between interface and VILLASnode, the protocol conversion time is approximately 0.15 ms.

TABLE 4. Summary of delay contributions.

Average time from NW interface to VILLASnode	0.085 ms
Average NW interface time	0.07 ms
Average protocol conversion time	0.15 ms

D. SUMMARY OF DELAY CONTRIBUTIONS

Table 4 summarizes the results. It shows that the virtual gateway VILLASnode and the network interface delays are very small compared to average internet latencies of 10 ms. They influence the transmission delay by maximal 3%. It also means that the architecture has a minor impact on the choice of application of the remote component testing. The main impact is still the internet latency. This impact can even be mitigated by using a higher setpoint update time. Depending on the specific requirements of the facilities, experiments can be designed so that the impact on the delay is minimized. Although the experiment is designed as realistic as possible, the requirements and measurements might change in real scenarios which requires future investigations.

IV. CASE STUDY

To verify the functionality of the architecture, an appropriate case study of three distributed facilities with relaxed time constraints is chosen. One facility serves as power generation (source), another one as load (sink). The third facility serves as remote grid to emulate both and to test the physical components in this remote grid by converters. The facilities are explained in more detail in the following sub-sections where the dynamic range as well as the implementation of VILLASnode is described.

A possible application is the integration of unavailable physical components in a microgrid to test primary control strategies based on the measurements of the physical components. For example, it can be understood as continuation of the case study in [9] where the microgrids are remotely distributed and can only communicate via VILLASnode.

A. INSTANCE OF ARCHITECTURE

For the case study, an instance of the architecture is used connecting three distributed facilities. The signal exchange is realized by sending power setpoints every 100 ms to 5 s. The exchange rate is defined by the control systems. The internet latency was in range of 10 ms measured by a Ping test. Since the exchange rate and the internet latency is very high compared to the previous calculated delay contributions of the architecture, these delays are neglectable. The delay

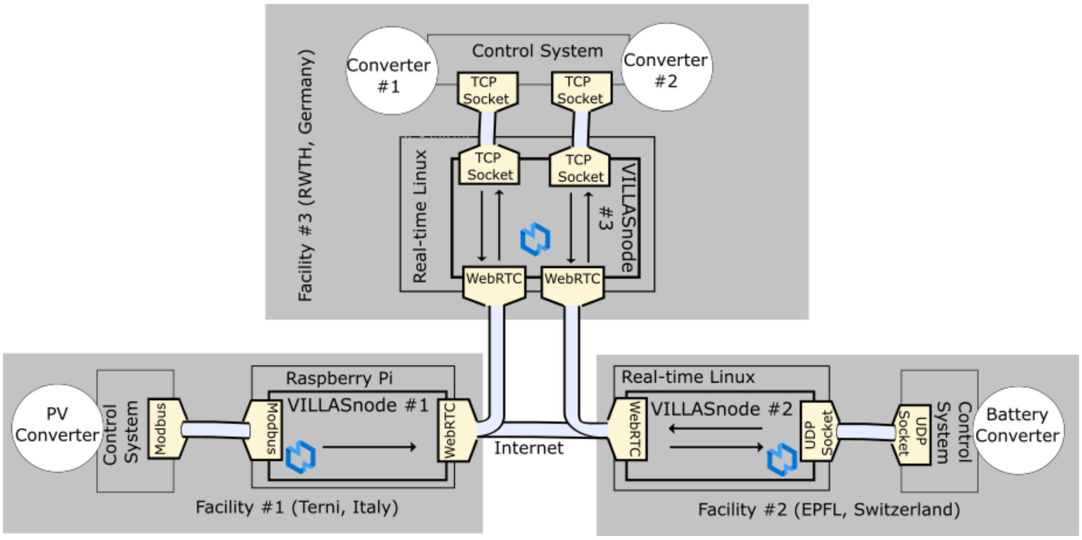

FIGURE 6. Architecture for case study.

impact of the virtual gateway, network interfaces, and internet is reduced to less than 1%. In this case, the components behave as if they are in one facility so that we can speak of a real-time experiment.

Figure 6 shows in detail the case study. The three gray areas represent the distributed facilities. In each facility, there is a control system which VILLASnode interfaces and gives the access to the converters. Two converters in facility 1 and 2 serve as physical component, whereas facility 3 has the remote converters to emulate the power setpoints. The three facilities are interconnected via the internet using WebRTC. In this way, PV and battery are tested remotely in facility 3 as well as the grid of facility three can be tested with different power generations or loads. The next sub-sections describe in detail the different facilities which are used to verify the architecture in operation.

B. FACILITY #1: ASM TERNI S.p.a

In Terni, monocrystalline solar panels are installed: two strings in parallel, each composed of 18 panels in series. Each panel delivers a voltage of 34.3 V at maximum power, i.e., at the highest level of solar irradiation. At open circuit, i.e., when the panel terminals are not connected to a load, the voltage is 41.2 V. The DC/DC converter has a voltage of $18 \times 34.3 = 617.4$ V. The total peak power output of the photovoltaic system with the two strings in parallel is around 12 kW. The PV system uses the 40kW / 750V Redprime ZEKALABS power DC/DC converter with an air-cooled unit. The PV converter voltage and current measurements go to a control system. VILLASnode can interface this control system by Modbus. The control systems sends out the measurements every 100 ms. VILLASnode forwards the measurements to EPFL and RWTH via the internet using WebRTC.

C. FACILITY #2: EPFL

The École Polytechnique Fédérale de Lausanne (EPFL) facility uses a Battery Energy Storage System (BESS) consisting of a commercial lithium-ion battery. It is produced by Samsung with a capacity of 28 kWh/ 30 kW. The BESS consists of 9 modules that are connected in series to create a voltage of around 700 V. It communicates with a resource agent. The agent is responsible for the communication with the battery through Modbus and the start-up, shut down, and error handling process. A central controller computes setpoints every 5 seconds for BESS and RWTH based on the implemented grid control strategy, which is load profile or droop control on the state of charge (SoC) of BESS. The central controller sends the setpoints to the resource agent. The central controller also serves as an interface for the virtual coupling via VILLASnode.

Droop control is implemented according to the difference of the SoC and a droop coefficient K_V:

$$P_{batt} = K_v \cdot \left(SoC_{ref} - SoC\right)$$

The droop control strategy ensures grid stability by balancing power generation and consumption dynamically, using the difference between the battery's SoC and a reference value. It challenges the architecture by introducing complex dependencies, as real-time remote measurements of PV power and battery consumption are critical to calculate and adjust setpoints accurately. This increases the need for robust synchronization and low-latency communication across facilities.

The load profile strategy provides precise control over energy usage by simulating predefined consumption patterns. It challenges the architecture by requiring accurate scaling and validation of power setpoints to prevent exceeding microgrid limitations of the third facility. Additionally, it tests

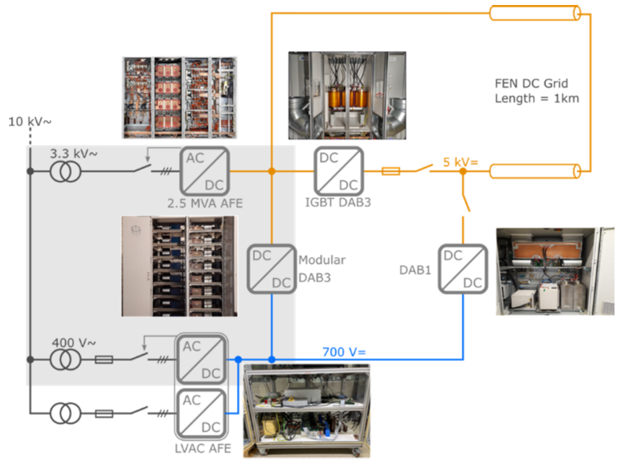

FIGURE 7. RWTH remote grid.

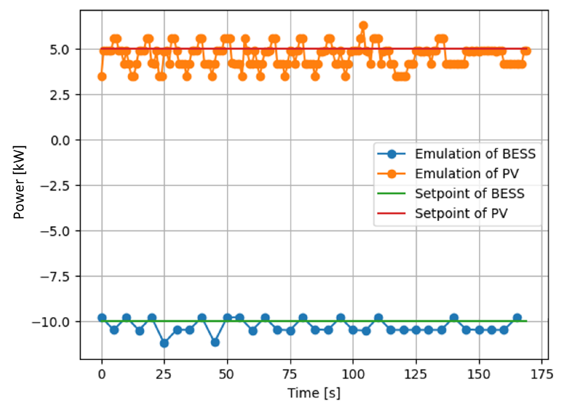

FIGURE 8. Droop control.

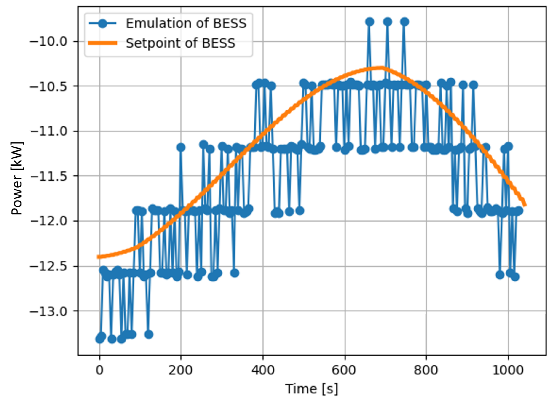

FIGURE 9. Load profile.

the system's ability to process and emulate large and dynamic power ranges while maintaining safe and reliable operation.

D. FACILITY #3: PGS (RWTH)

The German facility consists of two AC/DC converters (50 kW), built by the Institute for Power Generation and Storage Systems (PGS) of RWTH, and a constant DC voltage source (15 kW). Figure 7 shows the involved equipment in the gray area, whereas the rest shows the general wiring of the facility. Both AC/DC converters emulate the corresponding physical components from Italy and Switzerland. They are connected by a CAN-bus to Beckhoff CX7050 embedded computers. It is the interface with the so-called grid server, the control system in place. More information is available in [10]. VILLASnode talks with the grid server. It receives the power setpoints from Terni and EPFL, forwards them to the grid server so that the converters emulate them. To control the emulation and the reaction of the grid to the emulation, current and voltage of the converters are measured and send every 100 ms to the grid server forwarding it to VILLASnode.

E. APPLICATION: EMULATION OF MISSING GRID COMPONENTS WITH REMOTE CONTROL STRATEGIES

This application shows a possibility how to use the architecture and demonstrates its functionality. In an available microgrid (facility 3) without PV and battery, these missing components are emulated to test the microgrid with the dynamics of this load and generation as if the missing components were physically present. This capability is particularly important for validating control strategies, such as droop control or dynamic load profiles, under realistic conditions where distributed inputs are integrated.

Droop control was the first control strategy. Figure 8 shows power setpoints in kilowatts over the time in seconds. Based on the color, it is differentiated between the received setpoints of PV and battery and how they were emulated in German facility. They follow the incoming setpoints. However, some ripples can be seen which are caused by the systems voltage and current resolutions in one volt or one ampere steps. Consequently, when multiplying current and voltage to obtain the power, this error gets larger. This error can be reduced when increasing the power range, which was not possible during the experiments because of the power bus limitations of the microgrid.

The second control strategy was the following of a load profile. Figure 9 shows the power setpoints in kilowatts over time in seconds. The converters and thus the available DC microgrid follow the load profile constrained again by the resolution. This is not constrained by the architecture itself because it is designed to forward the values received from the underlying system and not to take care of resolution or frequency of updates.

F. DISCUSSION

The case study showcases the emulation of missing microgrid components using remote control strategies. This physcial-component-only approach enables collaborative testing of PV generation and battery storage across geographically distributed facilities. Although the converters were not specifically optimized for this emulation, the architecture supports control strategies such as droop control and dynamic load profiles. This indicates that any converter capable of implementing these system-level control strategies can be a viable use case, regardless of its hardware configuration.

Delays from the VILLASnode architecture were negligible compared to the internet latency (around 10 ms), which ensures real-time behavior of the system. This establishes a performance boundary that highlights the effectiveness of the communication setup in maintaining responsiveness, independent of VILLASnode itself. It emphasizes that remote emulation is feasible for certain systems, suggesting that this approach can be generalized beyond just specific architectures or protocols.

This case study has potential for generalization due to its modular architecture. The modularity allows for the replacement of physical components with compatible alternatives, such as different converters or energy storage systems that can interface with VILLASnode. This supports the extension of the architecture to more advanced case studies, including additional renewable energy sources and varying load profiles.

Moreover, the architecture can adapt to different time constraints, balancing current practical needs with the demands of future, more complex scenarios. This adaptability ensures it remains relevant as technology evolves and grid requirements change.

Future case studies may include, e.g., electric vehicle (EV) charging stations. An EV (or just the battery with the control) would be connected to a converter which emulates the charging station. The charging station would be in a different place and has a dynamic load. Both need to be equipped by VILLASnode to measure the power to (dis-)charge and to transmit the measurements. In this setup, different charging profiles or different battery control implementations can be tested with real measurements which would not be possible in one single facility.

V. CONCLUSION

This paper introduces a new approach to testing physical components in remote grids. It positions the approach as a complementary method to traditional real-time simulations. The proposed architecture is lightweight, making it easy to install and operate. The key feature of this architecture is the use of converters to emulate remote physical components and VILLASnode, which acts as a virtual gateway for data exchange between distinct facilities. This design choice allows for seamless integration with existing infrastructure without requiring significant modifications.

One of the findings of this study is that WebRTC serves as a effective communication protocol for facilitating remote connections between distinct locations. Unlike other protocols such as UDP combined with a VPN, WebRTC reduces user interaction during the setup phase. It reduces the setup time and avoids errors by automating many processes and thus enhancing usability. The practical advantage is that less manual interaction is required which makes the communication setup more user-friendly.

Furthermore, a performance comparison between UDP and WebRTC revealed that WebRTC outperforms UDP, especially in environments with noisy channels. While WebRTC maintains the original latency, UDP introduces additional delays. It makes WebRTC a more reliable choice for distributed experiments that involve remote physical components. However, the study also notes that WebRTC's performance may degrade if the data exchange must go through a relay server. This highlights the importance of carefully assessing the network setup to choose the optimal protocol.

The delay analysis conducted as part of this paper provided valuable insights into the time contributions of various components in the system. Specifically, the architecture was found to require 0.42 ms for protocol conversion, 0.17 ms from the network interface, and 0.15 ms for the actual conversion time. These breakdowns help identify areas where further optimizations could be made to minimize latency and improve overall system efficiency. The delay analysis also has practical implications to be used to fine-tune the setup for specific experiments. For example, understanding the time taken for protocol conversion and network interface communication can help optimize these components to minimize delays, which is critical when working with more time-sensitive remote testing.

A key application of this architecture was demonstrated in a case study, where the system was tested across various grids and control systems. This case study confirmed the system's flexibility, showing that it could operate effectively on diverse setups. Although the experiment did not require particularly fast update times, the delay analysis indicated the potential for pushing the limits of the system for future experiments that might demand lower latency and more precise timing.

A. FUTURE WORK

Future work might address the mentioned limitations of the architecture. This can included the investigation of a tool to facilitate the managing of large-scale experiments. Another direction could be to make the architecture more reliable for poor network conditions.

Generally, since this paper could show the communication aspect of the architecture, future experiments can focus more on advanced case studies. For instance, the setpoints of the physical component and the remote grid can serve as input to a digital twin architecture.

Another possibility would be an application in Energy internet (EI). One very important part of EI is the communication infrastructure which could build on VILLASnode. Because of its plug-and-play characteristic, it fits very well to the concept of EI. Components from different vendors could be integrated independently and communicate via VILLASnode.

ACKNOWLEDGMENT

The authors would like to thank the partners responsible of the different facilities to provide the equipment and infrastructure for the experiment.

REFERENCES

[1] M. H. Syed, E. Guillo-Sansano, Y. Wang, S. Vogel, P. Palensky, G. M. Burt, Y. Xu, A. Monti, and R. Hovsapian, "Real-time coupling of geographically distributed research infrastructures: Taxonomy, overview, and real-world smart grid applications," *IEEE Trans. Smart Grid*, vol. 12, no. 2, pp. 1747–1760, Mar. 2021, doi: 10.1109/TSG.2020.3033070.

[2] A. Monti, M. Stevic, S. Vogel, R. W. De Doncker, E. Bompard, A. Estebsari, F. Profumo, R. Hovsapian, M. Mohanpurkar, J. D. Flicker, V. Gevorgian, S. Suryanarayanan, A. K. Srivastava, and A. Benigni, "A global real-time superlab: Enabling high penetration of power electronics in the electric grid," *IEEE Power Electron. Mag.*, vol. 5, no. 3, pp. 35–44, Sep. 2018, doi: 10.1109/MPEL.2018.2850698.

[3] VILLASframework. *VILLASnode*. Accessed: Oct. 29, 2024. [Online]. Available: https://villas.fein-aachen.org/docs/node/

[4] K. Ma, J. Wang, X. Cai, and F. Blaabjerg, "AC grid emulations for advanced testing of grid-connected converters—An overview," *IEEE Trans. Power Electron.*, vol. 36, no. 2, pp. 1626–1645, Feb. 2021, doi: 10.1109/TPEL.2020.3011176.

[5] G. Perna, D. Markudova, M. Trevisan, P. Garza, M. Meo, M. M. Munafò, and G. Carofiglio, "Real-time classification of real-time communications," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 4, pp. 4676–4690, Jul. 2022, doi: 10.1109/TNSM.2022.3189628.

[6] S. Vogel, F. Wege, and A. Monti, "WebRTC-based plug-and-play signal transport for peer-to-peer connectivity between DRTSs, IEDs and operators," in *Proc. NEIS Conf. Sustain. Energy Supply Energy Storage Syst.*, 2024, pp. 1–12.

[7] E. Guillo-Sansano, M. H. Syed, A. J. Roscoe, G. M. Burt, and F. Coffele, "Characterization of time delay in power hardware in the loop setups," *IEEE Trans. Ind. Electron.*, vol. 68, no. 3, pp. 2703–2713, Mar. 2021, doi: 10.1109/TIE.2020.2972454.

[8] M. Pitz, F. Wege, N. Eiling, S. Vogel, V. Bareis, and A. Monti, "Automated deployment of single-board computer based grid measurement and co-simulation equipment," in *Proc. Open Source Model. Simul. Energy Syst. (OSMSES)*, Vienna, Austria, Sep. 2024, pp. 1–6, doi: 10.1109/osmses62085.2024.10668996.

[9] R. Bhattarai, J. Qi, J. Wang, and S. Kamalasadan, "Adaptive droop control of coupled microgrids for enhanced power sharing and small-signal stability," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, Aug. 2020, pp. 1–5, doi: 10.1109/PESGM41954.2020.9281797.

[10] J. Mathé and K. Hetzenecker, 2024, "Report on the demo site grid setups (1.7)," HYPERRIDE Consortium, doi: 10.5281/zenodo.14262702.

ALEXANDRA BACH received the B.Sc. degree in electrical engineering and the M.Sc. degree in information technology and computer engineering from RWTH Aachen University, in 2020 and 2023, respectively, where she is currently pursuing the Ph.D. degree with the E.ON Energy Research Center, Institute for Automation of Complex Power Systems. Her research interests include distributed real-time simulation and the automation of communication setups.

ANTONELLO MONTI (Senior Member, IEEE) received the M.Sc. (summa cum laude) and Ph.D. degrees in electrical engineering from the Politecnico di Milano, Milan, Italy, in 1989 and 1994, respectively. He started his career with Ansaldo Industria and then moved to the Politecnico di Milano, in 1995, as an Assistant Professor. In 2000, he joined the Department of Electrical Engineering, University of South Carolina, Columbia, SC, USA, as an Associate Professor and later became a Full Professor. Since 2008, he has been the Director of the Institute for Automation of Complex Power System, E.ON Energy Research Center, RWTH Aachen University, Aachen, Germany. Since 2019, he also has a joined appointment with the Fraunhofer FIT within the Center for Digital Energy Aachen. He has authored or co-authored more than 400 peer-reviewed papers published in international journals and the proceedings of international conferences. He was a recipient of the 2017 IEEE Innovation in Societal Infrastructure Award. He is an Associate Editor of *IEEE Electrification Magazine*. He is a member of the editorial board of the *Sustainable Energy, Grids and Networks*; and a member of the founding board of the *Energy Informatics*.